TO: Examiner Tecklu

Ph: (571) 272-7957

FAX: (571) 273-7957

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: **Mark Nathan Wain, et al.**	Confirmation No.: **1285**
Application No.: **10/678,870**	Group Art Unit: **2192**
Filing Date: **October 3, 2003**	Examiner: **Isaac Tuku Tecklu**

For: **Extensible Multi-Language Compilation**

REQUEST FOR EXAMINER INTERVIEW

Dear Examiner,

Thank you for scheduling an interview for Tuesday, March 31, 2009 at 3:00 P.M. EST. Below is a revised proposed claim response.

Sincerely,

Tom Van Zandt, 43,219

Phone: (408) 391-8489

Email: tavanzandt@earthlink.net

Listing of Proposed Amendments to Independent Claims:

Listing of Claims:

1. (Currently amended) A method for multi-language compilation comprising:

discovering a first compiler component that conforms to a set of interfaces included in ~~which implements~~ a prespecified protocol and provides compilation of code in a first language;

discovering a second compiler component that conforms to the set of interfaces included in ~~which implements~~ said prespecified protocol and provides compilation of code in a second language; and

providing a first framework which uses said first compiler component and said second compiler component to provide multi-language compilation, wherein the framework integrates the first compiler component and the second compiler component to create a multi-language compiler.

13. (Currently amended) A method for multi-language code editing comprising:

discovering a first design tool that conforms to a set of interfaces included in ~~which implements~~ a prespecified protocol and provides design assistance for code in a first language;

discovering a second design tool that conforms to the set of interfaces included in the ~~which implements a~~ prespecified protocol and provides design assistance for code in a second language; and

providing a framework which integrates said first design tool and said second design tool to provide multi-language editing, wherein the framework creates a multi-language design tool based on the first language and the second language.

14. (Currently amended) A method for multi-language code editing comprising:

discovering a first compiler component that conforms to a set of interfaces included in ~~which implements~~ a prespecified protocol and provides compilation of code in a first language;

discovering a second compiler component that conforms to the set of interfaces included in ~~which implements~~ said prespecified protocol and provides compilation of code in a second language;

discovering a first design tool that conforms to the set of interfaces included in the ~~which implements a~~ prespecified protocol and provides design assistance for code in said first language;

discovering a second design tool that conforms to the set of interfaces included in the ~~which implements a~~ prespecified protocol and provides design assistance for code in said second language; and

providing a framework which integrates said first compiler component, said second compiler component, said first design tool, and said second design tool to provide multi-language design functionality assisting a user in writing code comprising inter-language reference requirements, wherein the framework creates a multi-language compiler and a multi-language design tool based on the first language and the second language.

15. (Currently amended) A method for multi-language code editing comprising:

discovering a first compiler component that conforms to a set of interfaces included in ~~which implements~~ a prespecified protocol and provides compilation of code in a first language;

discovering a second compiler component that conforms to the set of interfaces included in ~~which implements~~ said prespecified protocol and provides compilation of code in a second language;

discovering a first design tool that conforms to the set of interfaces included in the ~~which implements a~~ prespecified protocol and provides design assistance for code in said first language;

discovering a second design tool that conforms to the set of interfaces included in the ~~which implements a~~ prespecified protocol and provides design assistance for code in said second language; and

providing a framework which integrates said first compiler component, said second compiler component, said first design tool, and said second design tool to provide multi-language design functionality incorporating background compilation, wherein the framework creates a multi-language compiler and a multi-language design tool based on the first language and the second language.

16. (Currently amended) A method for multi-language compilation comprising:

discovering at least two compiler components that each conform to a set of interfaces included in ~~which each implement~~ a prespecified protocol, wherein each of the at least two compiler components provides compilation of code in a corresponding language; and

providing a framework which integrates each of said at least two compiler components to provide multi-language compilation, wherein the framework creates a multi-language compiler based on the corresponding languages of the at least two compiler components.

19. (Currently amended) A computer-readable storage medium for multi-language compilation comprising computer executable modules having computer executable instructions, said instructions comprising instructions for:

discovering a first compiler component that conforms to a set of interfaces included in ~~which implements~~ a prespecified protocol and provides compilation of code in a first language;

discovering a second compiler component that conforms to the set of interfaces included in ~~which implements~~ said prespecified protocol and provides compilation of code in a second language; and

providing a first framework which uses said first compiler component and said second compiler component to provide multi-language compilation, wherein the framework creates a multi-language compiler based on the first language and the second language.

31. (Currently amended) A computer-readable storage medium for multi-language code editing comprising computer executable modules having computer executable instructions, said instructions comprising instructions for:

discovering a first design tool that conforms to a set of interfaces included in

~~which implements~~ a prespecified protocol and provides design assistance for code in a first language;

discovering a second design tool that conforms to the set of interfaces included in the ~~which implements a~~ prespecified protocol and provides design assistance for code in a second language; and

providing a framework which integrates said first design tool and said second design tool to provide multi-language editing, wherein the framework creates a multi-language design tool based on the first language and the second language.

32. (Currently amended) A computer-readable storage medium for multi-language code editing comprising computer executable modules having computer executable instructions, said instructions comprising instructions for:

discovering a first compiler component that conforms to a set of interfaces included in ~~which implements~~ a prespecified protocol and provides compilation of code in a first language;

discovering a second compiler component that conforms to the set of interfaces included in ~~which implements~~ said prespecified protocol and provides compilation of code in a second language;

discovering a first design tool that conforms to the set of interfaces included in the ~~which implements a~~ prespecified protocol and provides design assistance for code in said first language;

discovering a second design tool that conforms to the set of interfaces included in the ~~which implements a~~ prespecified protocol and provides design assistance for code in said second language; and

providing a framework which integrates said first compiler component, said second compiler component, said first design tool, and said second design tool to provide multi-language design functionality assisting a user in writing code comprising inter-language reference requirements, wherein the framework creates a multi-language compiler and a multi-language design tool based on the first language and the second language.

33. (Currently amended) A computer-readable storage medium for multi-language code editing comprising computer executable modules having computer executable instructions, said instructions comprising instructions for:

discovering a first compiler component that conforms to a set of interfaces included in ~~which implements~~ a prespecified protocol and provides compilation of code in a first language;

discovering a second compiler component that conforms to the set of interfaces included in ~~which implements~~ said prespecified protocol and provides compilation of code in a second language;

discovering a first design tool that conforms to the set of interfaces included in the ~~which implements a~~ prespecified protocol and provides design assistance for code in said first language;

discovering a second design tool that conforms to the set of interfaces included in the ~~which implements a~~ prespecified protocol and provides design assistance for code in said second language; and

providing a framework which integrates said first compiler component, said second compiler component, said first design tool, and said second design tool to provide multi-language design functionality incorporating background compilation, wherein the framework creates a multi-language compiler and a multi-language design tool based on the first language and the second language.

34. (Currently amended) A computer-readable storage medium for multi-language code compilation comprising computer executable modules having computer executable instructions, said instructions comprising instructions for:

discovering at least two compiler components that each conform to a set of interfaces included in ~~which each implement~~ a prespecified protocol, wherein each of the at least two compiler components provides compilation of code in a corresponding language; and

providing a framework which integrates each of said at least two compiler components to provide multi-language compilation, wherein the framework creates a multi-language compiler based on the corresponding languages of the at least two compiler components.

REMARKS

Claims 1-36 are pending in the application. Claims 1, 13-17, 19, and 31-35 have been amended. Claims 37-45 have been previously canceled. No claims have been added with this amendment. Therefore, claims 1-36 will remain pending in the application after entry of the foregoing claim amendments. Support for the amendments is found in the specification, drawings, and claims as originally filed. Applicants respectfully submit that no new matter has been added.

Claim Rejections – 35 U.S.C § 102

Claims 1-7, 9-25 and 27-36 stand rejected under 35 U.S.C. § 102(e) as allegedly being anticipated by Venter et al., U.S. Pub. No. 20040194072 (hereinafter "Venter"). Applicants respectfully traverse the rejections. Although Applicants believe that the present claims patentably define over Venter, Applicants have amended the claims to further clarify the claimed subject matter.

The cited portions of Venter fail to disclose or suggest the specific combination of claim 1 as amended. For example, the cited portions of Venter fail to disclose or suggest that the framework provides multi-language compilation by integrating the first compiler component and the second compiler component to create a multi-language compiler, as now recited in claim 1. Specifically, Venter is directed to a method for multi-language compilation that allows two or more separately written compiler components to co-operate in the compilation of mixed language compilation units. Venter uses a base class library and compilers for specific languages that are written as extensions of these classes. See Venter, Abstract. The cited portions of Venter disclose the following.

> Referring to FIG. 1, a multi-language compilation system 100 in accordance with an aspect of the present invention is illustrated. The system 100 includes a primary parser component 110, a primary compiler component 120 and a common compiler infrastructure (CCI) backend component 130. As illustrated in FIG. 2, optionally, the system 100 can include a first secondary parser component 140.sub.1 through an Nth secondary parser component 140.sub.N, N being an integer greater than or equal to one, and/or, a first secondary compiler component 150.sub.1 through an Mth secondary compiler component 150.sub.M, M being an integer greater than or equal to one. The first secondary parser component 140.sub.1 through the Nth secondary parser component 140.sub.N can be referred to collectively as the secondary parser component(s) 140. Similarly, the first secondary compiler component 150.sub.1 through the Mth secondary compiler

> component 150.sub.M can be referred to collectively as the secondary compiler component(s) 150.

(Venter, ¶ 31)

In Venter, the compiler for a specific language is written as extensions of the base classes of the common compiler infrastructure (CCI). The CCI provides a common format that allows creation of customized compiler components. In Venter, the system includes a primary compiler component and one or more secondary compiler components each of which are separately written using the CCI. In contrast, the invention, as claimed, integrates a number of compiler components that conform to a set of interfaces of a prespecified protocol. The use of a prespecified protocol allows the user to configure compilation constraints. The cited portions of Venter fail to disclose the creation of a multi-language compiler by integrating compiler components conforming to a set of interfaces included in a prespecified protocol. Thus, the cited portions of Venter fail to disclose or suggest the features of claim 1.

Applicants respectfully submit therefore that claim 1 patentably defines over Venter. As claims 2-7 and 9-12 depend from claim 1, claims 2-7 and 9-12 are also patentably define over Venter at least by virtue of their dependence from claim 1.

The subject matter of claim 1 discussed above is similarly recited in independent claims 13-16, 19 and 31-34. Therefore, claims 13-16, 19 and 31-34 likewise patentably define over Venter. As claims 17-18, 20-25, 27-30, 35, and 36 depend from claims 16, 19, and 34, these claims also patentably define over Venter at least by virtue of their dependence from claims 16, 19, and 34.

Claim Rejections – 35 U.S.C § 103

Claims 8 and 26 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable as obvious over Venter in view of Alpern (US 7,107,578). These rejections are respectfully traversed.

As discussed above, claims 1 and 19 patentably define over Venter. Alpern was cited in the Office Action mailed January 8, 2009 (Office Action) as allegedly disclosing a method for multiple languages to interface with a single debugger client using a standard interface. See Office Action, page 10, ¶ 1. The cited portions of Alpern fail to remedy the deficiencies of Venter discussed above. In particular, Alpern also fails to disclose the creation of a multi-language compiler as claimed. Therefore, Applicants respectfully submit that claims 1 and 19 patentably define over any proposed combination of the teachings of Venter and Alpern by one skilled in the art. As claims 8 and 26 depend from claims 1 and 19, these claims also patentably define over the combination of Venter and Alpern at least by virtue of their dependence from claims 1 and 19.

Accordingly, for at least the foregoing reasons, Applicants respectfully request withdrawal of the rejections of claims 1-36 under 35 U.S.C. §§ 102(e) and 103(a).